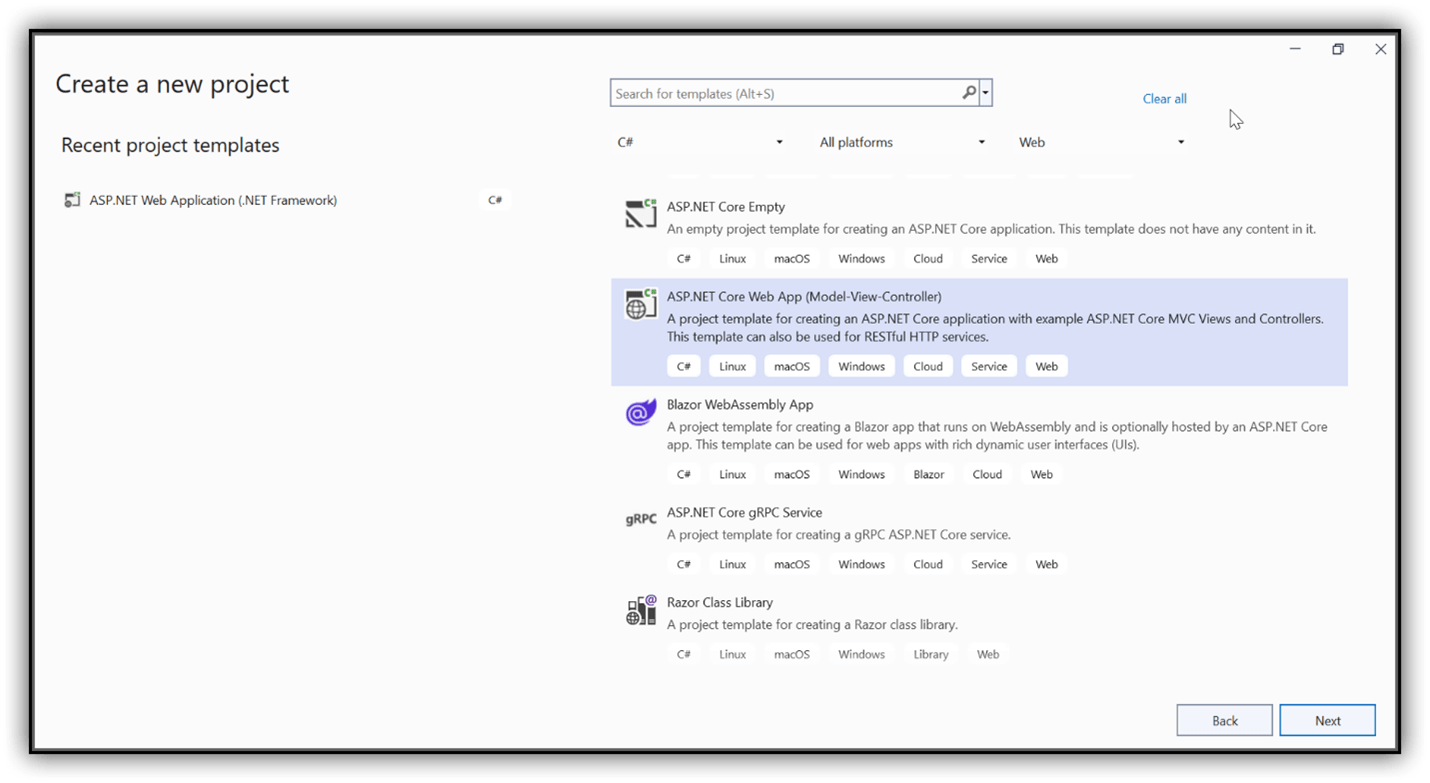
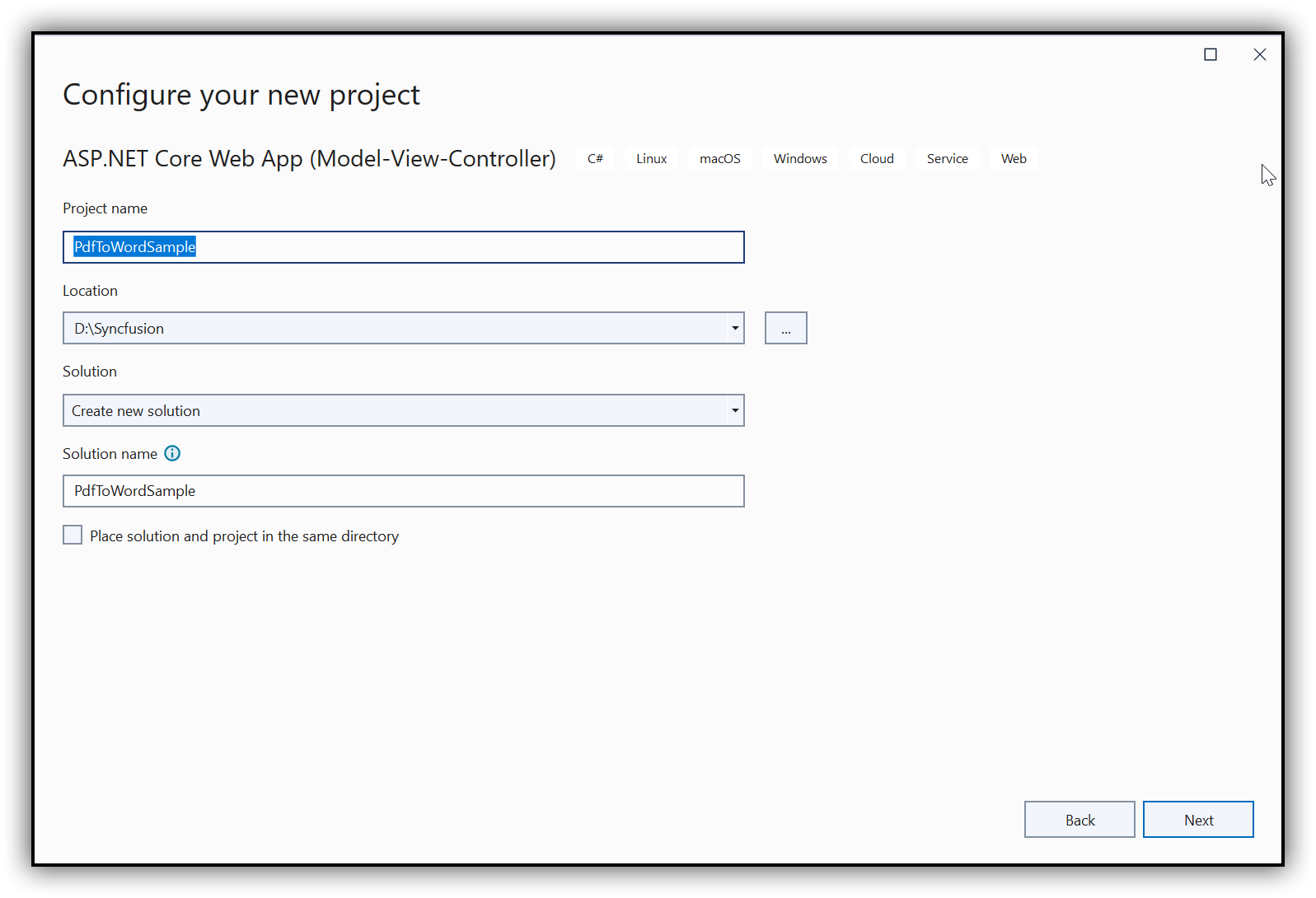
**Convert PDFs to Word Documents in ASP.NET Core**

Essential® [ASP.NET Core](https://www.syncfusion.com/document-processing/pdf-framework/net-core) PDF does not natively support converting PDF documents to Word format. However, as a workaround, you can convert PDF pages to images and then insert those images into a Word document using the DocIO library.

**Steps to Convert PDF to Word**:

1**. Set Up Your Project**: Initiate a new ASP.NET Core web application project.

2. Name your project on the configuration page and click Next.

A screenshot of a computer

AI-generated content may be incorrect.

3.**Install Necessary NuGet Packages**: Include [Syncfusion.EJ2.PdfViewer.AspNet.Core](https://www.nuget.org/packages/Syncfusion.EJ2.PdfViewer.AspNet.Core/) and [Syncfusion.DocIORenderer.Net.Core](https://www.nuget.org/packages/Syncfusion.DocIORenderer.Net.Core/) packages into your project from [nuget.org](https://www.nuget.org/).

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**4.** **Add a Conversion Button in View:** In your **index.cshtml**, create a form to trigger the conversion:

**CSHTML**

|  |
| --- |
| <div class="btn">  @{  Html.BeginForm("ExportToPDF", "Home", FormMethod.Post);  {  <input type="submit" value="Export To PDF" class="btn" />  }  }  </div> |

5. **Include Required Namespaces in Controller**: In **HomeController.cs**, add the following namespaces for PDF and Word manipulation:

**C#**

|  |
| --- |
| using Microsoft.AspNetCore.Mvc;  using System.Drawing;  using Syncfusion.DocIO.DLS;  using Syncfusion.Pdf.Parsing;  using Syncfusion.EJ2.PdfViewer;  using SkiaSharp; |
|  |

**VB.NET**

|  |
| --- |
| **Imports** System.Drawing  **Imports** Syncfusion.DocIO.DLS  **Imports** Syncfusion.Pdf.Parsing  **Imports** Syncfusion.EJ2.PdfViewer  **Imports** SkiaSharp  **Imports** Microsoft.AspNetCore.Mvc |

 6**. Implement Conversion Logic**: Define the **ExportToPDF** method in **HomeController.cs**:

**C#**

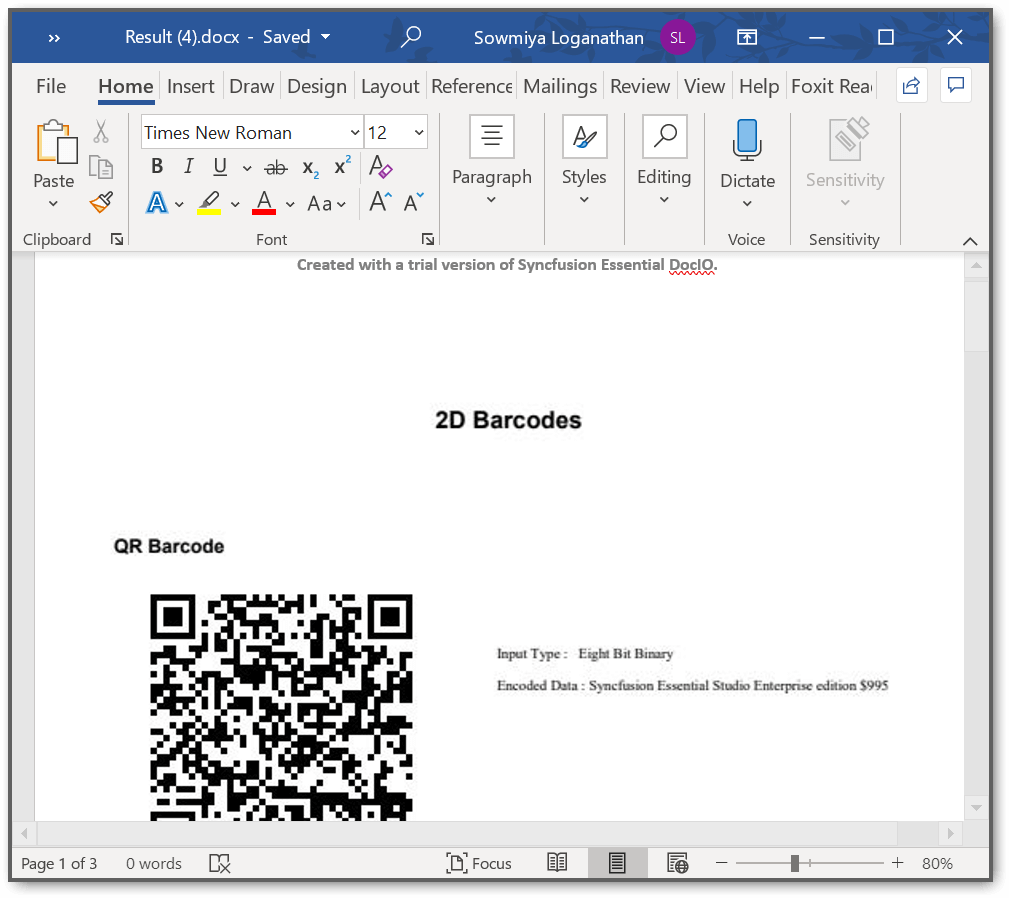
|  |
| --- |
| // Create a new Word document  using (WordDocument wordDocument = new WordDocument())  {  // Add a new section to the document  IWSection section = wordDocument.AddSection();  // Set the page margins to zero  section.PageSetup.Margins.All = 0;  // Add a new paragraph to the section  IWParagraph firstParagraph = section.AddParagraph();  SizeF defaultPageSize = new SizeF(  wordDocument.LastSection.PageSetup.PageSize.Width,  wordDocument.LastSection.PageSetup.PageSize.Height  );  string path = "Barcode.pdf";  // Open the PDF file stream  using (FileStream docStream = new FileStream(path, FileMode.Open, FileAccess.Read))  {  // Load the PDF document  using (PdfLoadedDocument loadedDocument = new PdfLoadedDocument(docStream))  {  PdfRenderer pdfExportImage = new PdfRenderer();  // Load the PDF into the renderer  pdfExportImage.Load(docStream);  for (int i = 0; i < loadedDocument.Pages.Count; i++)  {  // Export the PDF page as an image  using (SKBitmap bitmap = pdfExportImage.ExportAsImage(i))  using (SKImage image = SKImage.FromBitmap(bitmap))  using (SKData encodedData = image.Encode(SKEncodedImageFormat.Jpeg, 100))  using (MemoryStream imageStream = new MemoryStream())  {  encodedData.SaveTo(imageStream);  // Add the image to the Word document  IWPicture picture = firstParagraph.AppendPicture(imageStream);  picture.Width = defaultPageSize.Width;  picture.Height = defaultPageSize.Height;  }  }  // Save the Word document to a memory stream  using (MemoryStream stream = new MemoryStream())  {  wordDocument.Save(stream, Syncfusion.DocIO.FormatType.Docx);  stream.Position = 0;  // Return the Word file as a download  FileStreamResult fileStreamResult = new FileStreamResult(stream, "application/msword")  {  FileDownloadName = "Result.docx"  };  return fileStreamResult;  }  }  }  } |

VB.NET

|  |
| --- |
| ' Create a new Word document  Using wordDocument As New WordDocument()  ' Add a new section to the document  Dim section As IWSection = wordDocument.AddSection()  ' Set the page margins to zero  section.PageSetup.Margins.All = 0  ' Add a new paragraph to the section  Dim firstParagraph As IWParagraph = section.AddParagraph()  Dim defaultPageSize As New SizeF(  wordDocument.LastSection.PageSetup.PageSize.Width,  wordDocument.LastSection.PageSetup.PageSize.Height  )  Dim path As String = "Barcode.pdf"  ' Open the PDF file stream  Using docStream As New FileStream(path, FileMode.Open, FileAccess.Read)  ' Load the PDF document  Using loadedDocument As New PdfLoadedDocument(docStream)  Dim pdfExportImage As New PdfRenderer()  ' Load the PDF into the renderer  pdfExportImage.Load(docStream)  For i As Integer = 0 To loadedDocument.Pages.Count - 1  ' Export the PDF page as an image  Using bitmap As SKBitmap = pdfExportImage.ExportAsImage(i),  image As SKImage = SKImage.FromBitmap(bitmap),  encodedData As SKData = image.Encode(SKEncodedImageFormat.Jpeg, 100),  imageStream As New MemoryStream()  encodedData.SaveTo(imageStream)  ' Add the image to the Word document  Dim picture As IWPicture = firstParagraph.AppendPicture(imageStream)  picture.Width = defaultPageSize.Width  picture.Height = defaultPageSize.Height  End Using  Next  ' Save the Word document to a memory stream  Using stream As New MemoryStream()  wordDocument.Save(stream, Syncfusion.DocIO.FormatType.Docx)  stream.Position = 0  ' Return the Word file as a download  Dim fileStreamResult As New FileStreamResult(stream, "application/msword") With {  .FileDownloadName = "Result.docx"  }  Return fileStreamResult  End Using  End Using  End Using  End Using |

A complete working sample to convert PDF to Word in the ASP.NET Core platform can be downloaded from the [PDFToWordSample.zip](https://www.syncfusion.com/downloads/support/directtrac/general/ze/PdfToWordSample-83519269.zip).

By executing the sample, you will get the output document as follows.



Take a moment to peruse the [documentation](https://help.syncfusion.com/file-formats/docio/working-with-paragraph), where you can find the adding of text, images, lists, hyperlinks, and symbols in word paragraph. Explore more about the rich set of Syncfusion® [Word Framework](https://www.syncfusion.com/word-framework/net) features. An online example of [exporting PDF pages to image](https://ej2.syncfusion.com/aspnetcore/documentation/pdfviewer/how-to/export-as-image/)s.

Take a moment to explore the [documentation](https://help.syncfusion.com/file-formats/docio/working-with-paragraph), where you'll find guidance on adding text, images, lists, hyperlinks, and symbols to Word paragraphs using the Syncfusion® [Word Framework](https://www.syncfusion.com/word-framework/net). You can also discover a rich set of features supported by the framework, including an online example that demonstrates how to [export PDF pages to images](https://ej2.syncfusion.com/aspnetcore/documentation/pdfviewer/how-to/export-as-image/).

Note:

Starting with version **16.2.0.x**, if you're referencing Syncfusion® assemblies from the trial setup or the NuGet feed, you must include a valid license key in your projects. Click [here](https://help.syncfusion.com/common/essential-studio/licensing/license-key) to learn how to generate and register the Syncfusion® license key in your application to use the components without displaying a trial message.

**Conclusion**

We hope you found this guide helpful in learning how to convert PDF documents to Word format on the ASP.NET Core platform.

You can refer to our [**ASP.NET Core PDF feature tour**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/document-processing/pdf/pdf-library/net/create-pdf-file-in-asp-net-core), and how to quickly get started for configuration specifications. You can also explore our [**ASP.NET Core PDF example**](https://ej2.syncfusion.com/aspnetcore/PDF/Default#/bootstrap5) to understand how to create and manipulate data.

For current customers, you can check out our components from the [**License and Downloads**](https://www.syncfusion.com/sales/teamlicense) page. If you are new to Syncfusion; you can try our 30-day [**free trial**](https://www.syncfusion.com/downloads/aspnetcore-js2) to check out our other controls.

If you have any queries or require clarifications, please let us know in the comments section below. You can also contact us through our [**support forums**](https://www.syncfusion.com/forums), [**Direct-Trac**](https://support.syncfusion.com/create), or [**feedback portal**](https://www.syncfusion.com/feedback/aspnet-core?control=pdf). We are always happy to assist you!